ORF 555 HW 3

(Q.1) oil producer: $\frac{dx}{dt} = -q(x(t))$ as long as $x(t) > 0$

x = initial reserves $= x(0)$.

market price $= 1 - q(x(t))$

objective function $= J(q) = \int_0^{\tau_x^q} e^{-rt} q(x(t))(1 - q(x(t)))\,dt$

where τ_x^q is the exhaustion time from initial reserves x under the strategy q i.e. the time at which reserves run out.

(a) Time in yrs.

$90 = 100 e^{-r} \Rightarrow e^r = \frac{100}{90} = \frac{10}{9} \Rightarrow r = \ln\left(\frac{10}{9}\right)$

$\Rightarrow \boxed{r = 10.5361\%}$ $\Rightarrow \boxed{r = 0.105361}$

from here, $r = 0.1$

(b) $q = \frac{1}{2} \Rightarrow \frac{dx}{dt} = -\frac{1}{2}$

$x(0) = 10$

$$\Downarrow$$

$$x(t) - x(0) = -\frac{t}{2}$$

$$\Rightarrow x(t) = x(0) - \frac{t}{2} = 10 - \frac{t}{2}$$

~~t at which~~ $0 = x(\tau_x^q) = 10 - \frac{\tau_x^q}{2}$

$$\Rightarrow \boxed{\tau_x^q = 20 \text{ yrs}}$$

$$J(q) = \int_0^{20} e^{-0.1t} \frac{1}{2} \cdot \left(1 - \frac{1}{2}\right) dt$$

$$= \frac{1}{4} \int_0^{20} e^{-0.1t} dt$$

$$= \frac{1}{4 \cdot 0.1} \left[e^{-0.1t}\right]_{20}^{0}$$

$$= \frac{1}{0.4} \left(1 - e^{-2}\right)$$

$$= \frac{1}{0.4} (1 - 0.1353)$$

$$= \frac{0.8647}{0.4}$$

$$= \boxed{2.1617}$$

(c) $v(x) = \frac{1}{4r}\left(1 + W(-e^{-2rx-1})\right)^2$ (Eq^n(45) of Notes)

$v'(x) = \frac{1}{\cancel{4r}} \cdot \cancel{2} \cdot \left(1 + W(-e^{-2rx-1})\right)\left(W'(-e^{-2rx-1})\right)(+e^{-2rx-1})(+\cancel{2r})$

$= e^{-2rx-1}\, W'(-e^{-2rx-1})\left(1 + W(-e^{-2rx-1})\right)$

Let $y = W(x) \Rightarrow x = W^{-1}(y) = ye^y$

$\Rightarrow W(ye^y) = y$

$W'(x) = \frac{dW(x)}{dx} = \frac{dy}{dx}$

$\frac{1}{W'(x)} = \frac{dx}{dy} = \frac{d(ye^y)}{dy} = ye^y + e^y = x + \frac{x}{y} = \frac{xy + x}{y}$

$\Rightarrow W'(x) = \frac{y}{x(1+y)} = \frac{W(x)}{x(1+W(x))}$

At $t = \tau^{q^*}$, $x(t) = 0$, $\cancel{q(t) = 0 \Rightarrow v'(t) = 1}$

$q(0) = 0 \Leftarrow q(x(t)) = 0 \Rightarrow v'(x(t)) = 1 \Rightarrow v'(0) = 1$ ✓

$\left(\because q(x(t)) = \frac{(1 - v'(x(t)))}{2}\right)$ $(W(0) = 0)$

$\Rightarrow v'(x) = \cancel{e^{-2rx-1}}\, -\Delta\; W'(\Delta)\,(1 + W(\Delta))$

$= -\cancel{\Delta}\, \frac{W(\Delta)}{\cancel{\Delta}\,\cancel{(1+W(\Delta))}}\, \cancel{(1+W(\Delta))}$

$= -W(\Delta)$

$= \boxed{-W(-e^{-2rx-1})}$

$\Rightarrow v'(0) = -W(-e^{-1}) = -(-1) = 1$

$\Rightarrow q^*(x) = \frac{1 - v'(x)}{2} = \boxed{\frac{1 + W(-e^{-2rx-1})}{2}} \Rightarrow \boxed{v(x) = \frac{q^2(x)}{r}}$

Eq^n(43)

All in all, we have :-

$$\boxed{q^*(x) = \frac{1 + w(-e^{-2rx-1})}{2}} \quad \text{where } r = 0.1$$

~~$q^*(x) = 1 + w$~~

for time, we know that :-

$$v'(x(t)) = v'(x(0))\, e^{rt} \quad \text{(By Hotelling's rule)}$$

$$\Rightarrow v'(x(\tau)) = v'(x(0))\, e^{r\tau} \quad (\text{let } \tau = \tau_{10}^{q^*} \text{ of the } q^n)$$

$$\Rightarrow v'(0) = v'(10)\, e^{r\tau}$$

$$\Rightarrow \tau = \frac{1}{r} \ln\left(\frac{v'(0)}{v'(10)}\right) = \frac{1}{r} \ln\left(\frac{1}{v'(10)}\right)$$

$$= -\frac{\ln v'(10)}{r}$$

Now, $v'(x) = -w(-e^{-2rx-1})$

$$\Rightarrow v'(10) = -w(-e^{-2r\cdot 10-1}) = -w(-e^{-2-1}) = -w(-e^{-3})$$

$$\Rightarrow \boxed{\tau = \frac{-\ln(-w(-e^{-3}))}{r}}$$

$\Rightarrow$ From the computation, $\boxed{\tau = 29.4753}$ yrs

(d) $$v(x) = \frac{(1 + w(-e^{-2rx-1}))^2}{4r}$$

$$\Rightarrow v(10) = \frac{(1 + w(-e^{-2-1}))^2}{0.4} = \frac{(1 + w(-e^{-3}))^2}{0.4} = \boxed{2.2445}$$

~~$\Rightarrow v(10) = 2.2445$~~

(Q.2) P1 → oil P2 → solar, cost $c > 0$

$x(t)$ = oil reserves remaining at time t.

P1 → extract @ $q_1(x(t))$ P2 → extract @ $q_2(x(t))$

$$\frac{dx}{dt} = -q_1(x(t)) \text{ as long as } x(t) > 0$$

$q_1(0) = 0$

Price $= 1 - q_1 - q_2$

η = oil exhaustion time ⇒ $x(\eta) = 0$, $q(x(\eta)) = q(0) = 0$

P1 → $u(x) = \ldots$

P2 → $w(x) = \ldots$

$r = 0.1$

$x(0) = 10$

(a) cost → P1: $u'(x)$ | P2: $w'(x) + c$

$$q_1(x) = \frac{1 - u'(x)}{2}$$

Now, from proposition 4.4 (eq^n (5.9))

$$x_b = \frac{4}{9r}\left((1+c)\log\left(\frac{1+c}{2(2c-1)}\right) - 3(1-c)\right)$$

$$x_b = 5 \Rightarrow 5 = \frac{4}{0.9}\left((1+c)\log\left(\frac{1+c}{2(2c-1)}\right) - 3(1-c)\right)$$

$$\Rightarrow \frac{4.5}{4} = (1+c)\log\left(\frac{1+c}{4c-2}\right) - 3 + 3c$$

1.125

$$\Rightarrow 4.125 = (1+c)\log\left(\frac{1+c}{4c-2}\right) + 3c$$

(8)

So, to solve:-

$$\boxed{4.125 = (1+c)\log\left(\frac{1+c}{4c-2}\right) + 3c}$$

$$f(c) \to f'(c) = \left(\frac{\cancel{1+c}}{\cancel{1+c}}\right)\cancel{(4c-2)}\left(\frac{4c-2-(1+c)4}{(4c-2)^{\cancel{2}}}\right) + \log\left(\frac{1+c}{4c-2}\right) + 3$$

$$= \frac{\cancel{4c}-2-4-\cancel{4c}}{4c-2} + \log\left(\frac{1+c}{4c-2}\right) + 3$$

$$= -\frac{3}{2c-1} + 3 + \log\left(\frac{c+1}{4c-2}\right)$$

$$= 3 - \frac{3}{2c-1} + \log\left(\frac{1}{4}\left(\frac{4c+4}{4c-2}\right)\right)$$

$$= 3 - \frac{3}{2c-1} + \log\left(\frac{1}{4}\right) + \log\left(1 + \frac{6}{4c-2}\right)$$

$$= 3 - \frac{3}{2c-1} - \log 4 + \log\left(1 + \frac{3}{2c-1}\right)$$

At $c=1$, $f'(1) = 0$ (& $f(1) = 3$)

After solving using computational tools, we get:-

$\boxed{\cancel{c = 2.154}}$ $\boxed{c = 0.59}$ $(c < 1)$

(b) $u(x) = \frac{1}{4r}\left(1 + w\left(\theta(x - x_b)\right)\right)^2$ | $\theta(x) = (1-2c)e^{1-2c-2rx}$

$u(10) = \frac{1}{0.4}\left(1 + w\left(\theta(10-5)\right)\right)^2$ ∟ Eq^n (60)

$= 2.5\left(1 + w\left(\theta(5)\right)\right)^2$

$r = 0.1$ ↓

$\theta(x) = -e^{-2rx-1}$

(previous (Q1) case)

when $c = 1$

& $x_b = 0$

$\theta(5) = (1-2c)e^{1-2c-2r\times5}$

$= (1-2c)e^{\cancel{1}-2c-\cancel{1}}$ $(\because r = 0.1)$

$= (1-2c)e^{-2c}$

On solving computationally, we get :-

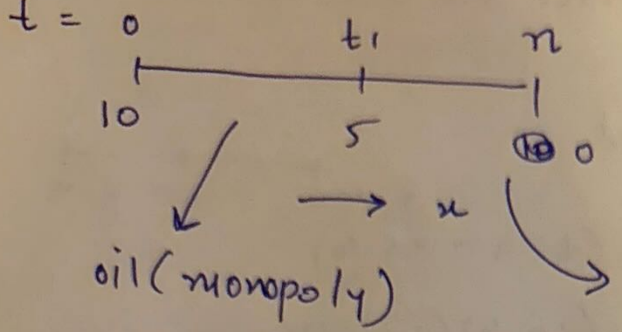
~~$\theta(5) = -0.0445$~~

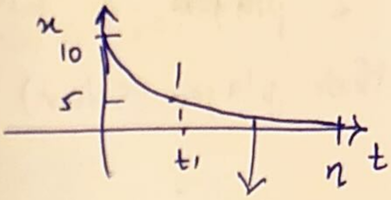
& ~~So, $u(10) = 2.27$~~

$\boxed{u(10) = 2.2139}$

(c) x goes from 10 to 0

i.e. $x(0) = 10$, $x(10) = 0$





$\left|\frac{dx}{dt}\right|_{t>t_1} < \left|\frac{dx}{dt}\right|_{t<t_1}$

(True otherwise also) - e when solar is not there)

So, from ~~to~~ $t=0$ to t_1, since there is a single player, Hotelling rule holds

$\Rightarrow v'(x(t)) = v'(x(0))\, e^{rt}$

$\Rightarrow v'(x(t_1)) = v'(x(0))\, e^{rt_1}$

$\Rightarrow v'(5) = v'(10)\, e^{rt_1}$ $\qquad v'(x) = -w(-e^{-2rx-1})$

$\Rightarrow t_1 = \frac{1}{r} \ln\left(\frac{v'(5)}{v'(10)}\right)$

$\cancel{=} \frac{1}{r} \ln\left(\frac{+w(-e^{-2r\cdot 5-1})}{+w(-e^{-2r\cdot 10-1})}\right)$

$= \frac{1}{r} \ln\left(\frac{w(-e^{-2})}{w(-e^{-3})}\right)$

$= \cancel{\boxed{11.06}}$

Here, $v'(x) \neq -w(-e^{-2rx-1}) \because c \neq 1$.

computed afterwards

Now, when $t > t_1$, solar comes into picture

$\Rightarrow$ 2 players $\Rightarrow$ But $\frac{dx}{dt} = -q_1(x(t))$ still since the other player (solar) does not extract oil ~~but just prov~~

Now, q_1^* has changed a bit

$t \rightarrow$ 0 —— t_1 —— η

Earlier (from q_1),

$$q_1^* = \frac{1 + w(\cdot)}{2}$$

$$= \frac{1 + w(-e^{-2rx-1})}{2}$$

Now ($t > t_1 \rightarrow x < x_b$)

$$q_1^* = (1 + w(\cdot))\left(\frac{1+c}{3}\right)$$

$$= \left(\frac{1+c}{3}\right)(1 + w(\cdot))$$

$$\cancel{= \frac{1.59}{3}\left(1 + w\left(-e^{-\frac{9rx}{4\times1.59}-1}\right)\right)}$$

$$= \left(\frac{1+c}{3}\right)\left(1 + w\left(-e^{-\frac{9rx}{4(1+c)}-1}\right)\right)$$

So, $\cancel{q_1^* = 1}$

By hotelling's rule,

$$u'(x(t_2)) = u'(x(t_1))\, e^{r(t_2-t_1)}$$

$$u'(x(\eta)) = u'(x(t_1))\, e^{r(\eta-t_1)}$$

$$u'(0) = u'(5)\, e^{r(\eta-t_1)}$$

$$\Rightarrow \eta - t_1 = \frac{1}{r}\ln\left(\frac{u'(0)}{u'(5)}\right)$$

Now, from eq^n (55),

$$u(x) = \frac{(1+c)^2}{9r}\left(1 + w(\theta(x))\right)^2 \quad \text{where} \quad \theta(x) = -\exp(-krx-1)$$

$$\& \; k = \frac{9}{4(1+c)}$$

Q.2(c) continued

$$\text{So, } u'(x) = \frac{(1+c)^2}{9r} \cdot 2\,(1+W(\theta(x)))\,W'(\theta(x))\,\theta'(x)$$

$$= \frac{(1+c)^2}{9r}\,(1+\cancel{W(\theta(x))})\,\frac{W(\theta(x))}{\theta(x)\,\cancel{(1+W(\theta(x)))}} \times \theta'(x)$$

$$= \frac{(1+c)^2}{9r}\,\frac{W(\theta(x))}{\cancel{\theta(x)}} \cdot \cancel{\theta(x)}\,(-kr) \qquad \because k = \frac{9}{4(1+c)}$$

$$= \frac{(1+c)^{\cancel{2}}}{\cancel{9r}}\,W(\theta(x))\,\left(\frac{-\cancel{9r}}{4\cancel{(1+c)}}\right)$$

$$= -\frac{(1+c)}{4}\,W(\theta(x))$$

$(\because \theta(x) < 0,\ W(\theta(x)) < 0 \Rightarrow u'(x) > 0$ which makes sense because as $x \uparrow$, $u'(x) \uparrow)$

$$\text{So, } \eta - t_1 = t_2 = \frac{L}{r} \ln\left(\frac{-\cancel{\frac{(1+c)}{4}}\,W(\theta(0))}{-\cancel{\frac{(1+c)}{4}}\,W(\theta(5))}\right)$$

$\Rightarrow \cancel{\boxed{t_2 = 0.37}}$ $\quad \boxed{t_2 = 14.80}$

shown afterwards

$\Rightarrow$ $\boxed{\text{Total time}}$ $= \eta = t_1 + t_2 = \overset{11.22}{\cancel{11.06}} + \cancel{0.37}\ 14.80$

$\cancel{= \boxed{11.43} = \boxed{25.86}}$ $= \boxed{26.02}$

Note :-

$\theta_{small}(x) = -e^{-\frac{9rx}{4(1+c)}-1}$, $\theta_{large}(x) = \cancel{e^{-2rx-1}}$

$(1-2c)e^{1-2c-2rx}$

~~For a given cost c,~~ & r fixed

Things to compare:-

For a given cost c, θ_{small} vs θ_{large}

(~~we are In both Q1 & Q2, from $x=t_0$ to $x(=x_b)=5$, only 1 player. However, in Q1, we use θ_{large}~~)

For θ_{small}, $c = \cancel{0.54}$ 1 vs $c = 0.59$

(Q1) (Q2)

Solar in action when $x < x_b$

oil will be consumed more slowly

$\Rightarrow t_2^{Q1} < t_2^{Q2}$

$\Rightarrow \eta^{Q1} < \eta^{Q2}$

$\boxed{\text{Duopoly}}$	$c > \frac{1}{2}$	$c < \frac{1}{2}$
$x > x_b$	Blockaded	NB
$x < x_b$	NB	NB

Consider $x \in (x_b, \infty) \Rightarrow$ Q1: Monopoly $\xrightarrow{c=1}$ θ_{large} with $c = 1$

Q2: Duopoly but since $x > x_b$ & $c = 0.59 > \frac{1}{2}$,

$\Rightarrow$ ~~Monopoly~~ $\Rightarrow$ θ_{large} with $c = 0.59$

Now, for $x > x_b \Rightarrow t \in [0, t_1]$

$$t_1 = \frac{1}{r} \ln\left(\frac{v'(5)}{v'(10)}\right) = \frac{1}{r} \ln\left(\frac{u'(5)}{u'(10)}\right)$$

$$\Rightarrow u(x) = \frac{1}{4r}\left(1 + W(\theta(x-5))\right)^2 \quad \text{where } \theta(x) = (1-2c) \times e^{1-2c-2rx}$$

$$u'(x) = \frac{1}{4r} 2 \left(1 + W(\theta(x-5))\right) W'(\theta(x-5))\, \theta'(x-5)$$

$$= \frac{1}{2r}\left(1 + W(\theta(x-5))\right) \frac{W(\theta(x-5))}{\theta(x-5)\left(1 + W(\theta(x-5))\right)} \theta'(x-5)$$

$$= \frac{1}{\cancel{2r}} * \frac{W(\theta(x-5))}{\theta(x-5)} * \left((1-2c)e^{1-2c} \cancel{(-2r)}\right) \theta(x-5)(-\cancel{2r})$$

$$= -W(\theta(x-5))$$

This θ is θ-large

$$\Rightarrow t_1 = \frac{1}{r} \ln\left(\frac{W(\theta(5-5))}{W(\theta(10-5))}\right)$$

$$= \frac{1}{r} \ln\left(\frac{W(\theta(0))}{W(\theta(5))}\right)$$

$$= \boxed{11.22}$$

Finally, $\boxed{\text{total time}} = \boxed{26.02}$ $(11.22 + 14.80)$

So, Monopoly $\Rightarrow$ Duopoly with $c = 1 \Rightarrow x_b = 0$

Monopoly	$c > \frac{1}{2}$	~~$c < \frac{1}{2}$~~
$x > x_b$	$B(\theta_l(c=1))$	
~~$x < x_b$~~		

Duopoly	$c > \frac{1}{2}$	$c < \frac{1}{2}$
$x > x_b$	$B(\theta_l(c))$	$NB(\theta_{small})$
$x < x_b$	$NB(\theta_{small})$	$NB(\theta_{small})$

Note :- $\theta_l(c=1)$ $\downarrow$ ~~11.06~~ $t_1 = 11.06$

$\theta_l(c = 0.59)$ $\downarrow$ $t_1 = 11.22$

Q_1: Monopoly

x:

$\to 5$ | t_1 with $\theta_l(c=1) \to 11.06$

$\to 0$ | t_2 with $\theta_l(c=1) \to 18.41$

$Q2$: Duopoly

t_1 with $\theta_l(c = 0.59) \to 11.22$

t_2 with $\theta_s(c = 0.59) \to 14.80$

$\Rightarrow$ As $c \downarrow$, $t_1 \uparrow$ (Not entirely true because we get 11.22 when arguments to θ_l from x side are $x - x_b$ i.e. 0 to 5

So, 11.22 should be compared with 18.41

$\Rightarrow$ As $c \downarrow$, $t_1 \downarrow$

Ideally, what should happen?

As $c \downarrow$, solar $\uparrow \Rightarrow$ player coming in but blockaded before x_b. So, it depends on how cost affects $q^*_{oil}(x)$

$\hat{}$ contain integral over $t \Rightarrow$ does not depend on t naively

Note that :-

★ $\boxed{v(x) = \frac{(1+w(\theta_{large}(1,r,x)))^2}{4r}}$

$\boxed{v'(x) = -w(\theta_1(1,r,x))}$

★ $\boxed{q^*(x) = \frac{1-v'(x)}{2} = \frac{1+w(\theta_1(1,r,x))}{2} = \sqrt{r\,v^*(x)}}$

★ $\boxed{\text{Exhaustion time } (\tau)}$

parameter of interest :-

① ~~when~~ optimal strategy $\to q^*(x(t))$

② Exhaustion time $\to \tau$

③ value earned till that $(t) \to v(x)$

In general,

when $x > x_b$ & $c > 1$, $v'(x) = -w(\theta_1(c,r,x))$

$$\Rightarrow q^*(x) = \frac{1+w(\theta_1(c,r,x_0-x_b))}{2}$$

~~As $c\uparrow$~~,

Note that $\theta_1(\;) = (1-2c)e^{1-2c-2rx}$

$= (1-2c)e^{1-2c}e^{-2rx}$ | $\theta_1(0,r,x) = e^{1}e^{-2rx}$, $\theta_1(1,r,x) = -e^{-1}e^{-2rx}$

$$\Rightarrow \frac{\partial\theta_1}{\partial c} = (1-2c)e^{1-2c}(-2) + e^{1-2c}(-2)$$

$= e^{1-2c} \times (-2)(1 \cancel{-2c} - 2c + 1)$

$= -2e^{1-2c} \cdot 2(1-c)$

$= -4(1-c)e^{1-2c}$

$< 0 \quad (\because c < 1)$

$\Rightarrow$ As $c\uparrow$, $\theta_1\downarrow \Rightarrow w(\theta_1)\downarrow$ ($\because w$ is $\uparrow$) $\Rightarrow q^*(x)\downarrow$

~~So, when~~

$$t_1 = \frac{1}{r}\ln\left(\frac{-v'(5)}{-v'(10)}\right)$$

$w'(\theta(x - x_b, 5))$ $= w'(\theta(0))$

$w'(\theta(5))$

has x_b which also depends on c!

So, when $c \to 1$, q^* is smallest.

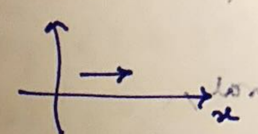
⇒ In monopoly, rate of extraction is the smallest

As the other player comes into picture $(c \downarrow)$,

$q^* \uparrow$ (& $x_b \uparrow$). ($c \to \frac{1}{2} \Rightarrow x_b \to \infty \Rightarrow$ Never blockaded)

~~(i) for $x > x_b$~~ as $c \downarrow$, $x_b \uparrow$

(i) for $x > x_b$, $q^*_{oil} \uparrow$



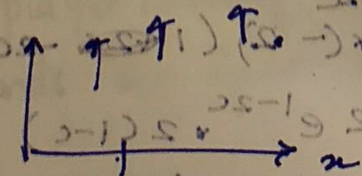
(ii) for $x < x_b$, $q^* = \dfrac{1 + w(\theta_s(c, r, x))}{2}$

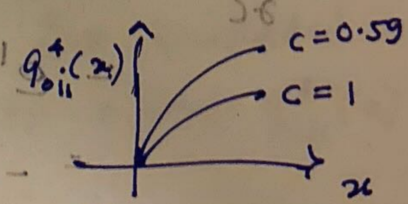
$$\theta_s(c, r, x) = -e^{-\frac{9rx}{4(1+c)} - 1}$$

$$\Rightarrow \frac{\partial \theta_s}{\partial c} = -e^{-\frac{9rx}{4(1+c)} - 1} \cdot \frac{9rx}{4(1+c)^2} < 0 \Rightarrow q^*_{oil} \uparrow$$

~~⇒ as $c \downarrow$, $\theta_s \uparrow$ ⇒ $q^*_{oil} \uparrow$~~

⇒ As $c \downarrow$, $x_b \uparrow$ & $\forall x$, $q^*_{oil} \uparrow$





⇓

$\tau_{0.59} < \tau_1$

(d) $q_1^{(0)*}(x) = \frac{1}{2}(1 + W(\theta(x - x_b)))$ | $q_2^{(0)*}(x) = 0$

$= \frac{1}{2}(1 + W(\theta(x - 5)))$ ↳ Eq^n 60

$\cancel{= \frac{1}{2}(1 + W(\theta(5)))}$

$= \frac{1}{2}(1 + W(\theta(x - 5)))$

The above is for $x \geq x_b$.

For $x < x_b$,

$q_1^{(0)*}(x) = \frac{(1+c)}{3}(1 + W(\theta(x)))$ | $q_2^{(0)*}(x) = \frac{1}{3}\left(1 - 2c - \frac{(1+c)}{2} W(\theta(x))\right)$

Eq^n 57 | Eq^n 58

Here $\theta(x) = -\exp\left(-\frac{grx}{4(1+c)} - 1\right)$

(e) After $t = \eta$, oil is exhausted $\Rightarrow q_1^*(x) = 0$

$\Rightarrow$ price $= \underline{\underline{1 - q_2^*(x(t))}}$.

$= 1 - q_2^*(0) \quad (\because x(t) = 0 \ \forall\, t \geq \eta)$

$= 1 - \left(\frac{1-c}{2}\right) = \frac{1+c}{2} = \frac{1.59}{2} = \boxed{0.795}$

Here, $x^*(t)$ = Nash Eq^{lb} when P1 is using $q_1^*(x)$

~~$x^*(t)$~~ Situation : $c > \frac{1}{2} \ \big| \ x \gtrless x_b$

$$\Rightarrow \frac{dx}{dt} = -q_1^*(x(t)) = -q_1^*(x)$$

$$\Rightarrow \frac{dx}{-q_1^*(x)} = dt$$

$$\int \frac{dx}{-q_1^*(x)} = dt$$

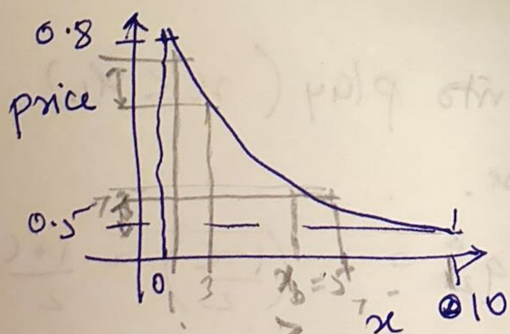
Roughly, as $t \uparrow (0 \to t_1 \to \eta)$, $x \downarrow$ & so, $q_1^*(x)$ will $\downarrow$

$\{ q_2^*(x)$ will $\uparrow$

But overall, $q_1^* + q_2^* \downarrow \Rightarrow$ price$(1 - q_1^* - q_2^*) \uparrow$

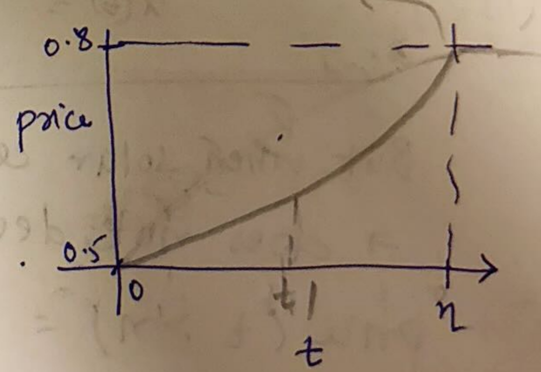
0.795 to be more precise

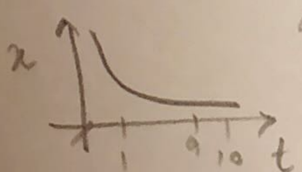
overall,



$\Delta P_2 > \Delta P_1$

$\Rightarrow$





$t \to x \to q_1^* \to P$
q_2^*

$\Delta = -3 \quad \Delta = -2$
$10 \to 7 \quad 3 \to 1$
$\Delta P_1 < \Delta P_2$

~~In our case~~. Ideally, after solar enters, we expect price to $\downarrow$ / atleast not $\uparrow$ too much

(continued on next page)

Mathematical way

In our case, (Lemma 4.2, 4.3 & around)

$$rv_0 = q_0(1-q_0-q_1-v_0') = g(v_0')$$

$$\Rightarrow g(x) = q_0(1-q_0-q_1-x)$$

$$g'(x) = -q_0$$

$$h(z) = \int_{g^{-1}(0)}^{z} \frac{g'(w)}{w}\,dw = \int_{g^{-1}(0)}^{z} -\frac{q_0}{w}\,dw = q_0 \ln\left(\frac{g^{-1}(0)}{z}\right)$$

$$\Rightarrow \frac{h(z)}{q_0} = \ln\left(\frac{g^{-1}(0)}{z}\right)$$

$$\Rightarrow e^{h(z)/q_0} = \frac{g^{-1}(0)}{z}$$

$$\Rightarrow z = g^{-1}(0)\exp\left(-\frac{h(z)}{q_0}\right) = \cancel{h^{-1}(z)}$$

$$\Rightarrow \boxed{h^{-1}(y) = g^{-1}(0)\exp\left(-\frac{y}{q_0}\right)}$$

So, $P(b) = 1 - q_1^*(x(t)) - q_2^*(x(t))$

$\cancel{=1-q^*}$

$O_{any}(x(t+s))$ $O_{m}(x)$

$$x(t) = \frac{1}{r} h(e^{rt} h^{-1} r x(0))$$

But when solar comes into play ($x < x_b$), price still ↑

→ does not decrease.

$$\text{price } (t > n) = 1 - q_2^* = 1 - \left(\frac{1-c}{2}\right) = \frac{1+c}{2} = 0.795 \sim 0.8$$

(consistent with the plot)

(Q. 4)

PJM

(C)

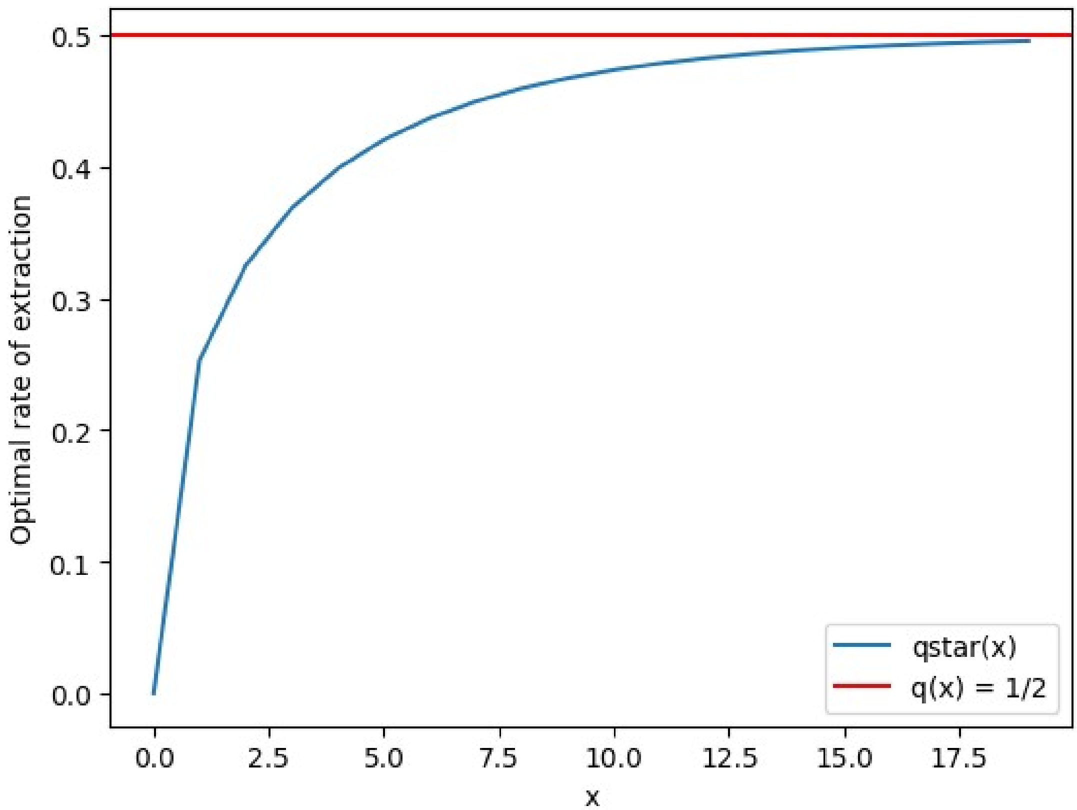
Type \ Time (AM) →	4	5	6	7	8
Forecast	75425	79257	85700	90678	91481
Actual	74840	76479	81793	88363	91411
Δ	585	2778	3907	2315	70
% (of forecast)	0.8	3.4	4.5	2.5	~0

```python
import math
from matplotlib import pyplot as plt
import numpy as np
from scipy.optimize import minimize
from scipy.special import lambertw
import pandas as pd

def mylambertw(x):
    if(x == -math.exp(-1)):
        return -1
    return lambertw(x)
```

Q1

(c)

```python
# xvec = []
# yvec = [0.1*i for i in range(-10, 30)]
# for y in yvec:
#     xvec.append(y*math.exp(y))
# plt.plot(xvec, yvec)
# plt.xlabel("x = yexp(y)")
# plt.ylabel("y")
# plt.title("W(x)")
# plt.show()

r = 0.1
def qstar(x):
    value = (1+mylambertw(-math.exp(-2*r*x-1)).real)/2
    return value

xvec = [i for i in range(0, 20)]
qstarvec = [] # Because lambertw(0) returns NaN when x=0
for x in xvec:
    qstarvec.append(qstar(x))
plt.plot(xvec, qstarvec, label = "qstar(x)")
plt.axhline(1/2, color = "r", label = "q(x) = 1/2")
plt.xlabel("x")
plt.ylabel("Optimal rate of extraction")
plt.legend()
plt.show()
```




```
print(mylambertw(-math.exp(-3)))
```

```
(-0.05246909745771487+0j)
```

```
tau = -math.log(-lambertw(-math.exp(-3)).real)/r
print(tau)
```

```
29.475309025422852
```

(d)

```
well_value = (1+lambertw(-math.exp(-3)).real)**2/0.4
print(well_value)
```

```
2.2445370281814934
```

Q2

```
r = 0.1
```

(a)

```
xb = 5

def f(c):
    return (1+c)*math.log((1+c)/(4*c-2)) + 3*c
```

```
def squared_error(pmt): #? How to pass args here?
    c = pmt[0]
    # return (f(c) - 4.125)**2
    return (4/(9*r)*((1+c)*math.log((1+c)/(2*(2*c-1))) - 3*(1-c)) -
xb)**2

output = minimize(squared_error, [0.55]) #? How to pass args here?

c = output.x[0]
print("cost:", c) # c has to be < 1; hence we start from below 1
```

```
cost: 0.5905573715778599
```

```
# cvec = [i for  i in range(1, 20)]
# fcvec = []
# for c in cvec:
#     fcvec.append(f(c))
# plt.plot(cvec, fcvec)
# plt.axhline(4.125, color='r')
# plt.show()
```

(b)

```
def theta_large(c, r, x):
    return (1-2*c)*math.exp(1-2*c-2*r*x)

def theta_small(c, r, x):
    return -math.exp(-9*r*x/(4*(1+c)) - 1)

def u(c, r, x, xb):
    return (1+mylambertw(theta_large(c, r, x-xb)).real)**2/(4*r)

well_value = u(c, r, 10, xb)
print(well_value)
```

```
2.213856275278776
```

(c)

```
t1 = 1/r*math.log(mylambertw(theta_large(c, 0.1,
0)).real/mylambertw(theta_large(c, 0.1, 5)).real)
print(t1)
```

```
11.22147425963023
```

```
t2 = 1/r*math.log(mylambertw(theta_small(c, 0.1,
0)).real/mylambertw(theta_small(c, 0.1, 5)).real) # eta - t1
print(t2)
```

```
14.795617563485173
```

Referencing Q1

```
t1 = 1/r*math.log(mylambertw(theta_large(1, 0.1,
5)).real/mylambertw(theta_large(1, 0.1, 10)).real)
print(t1)
```

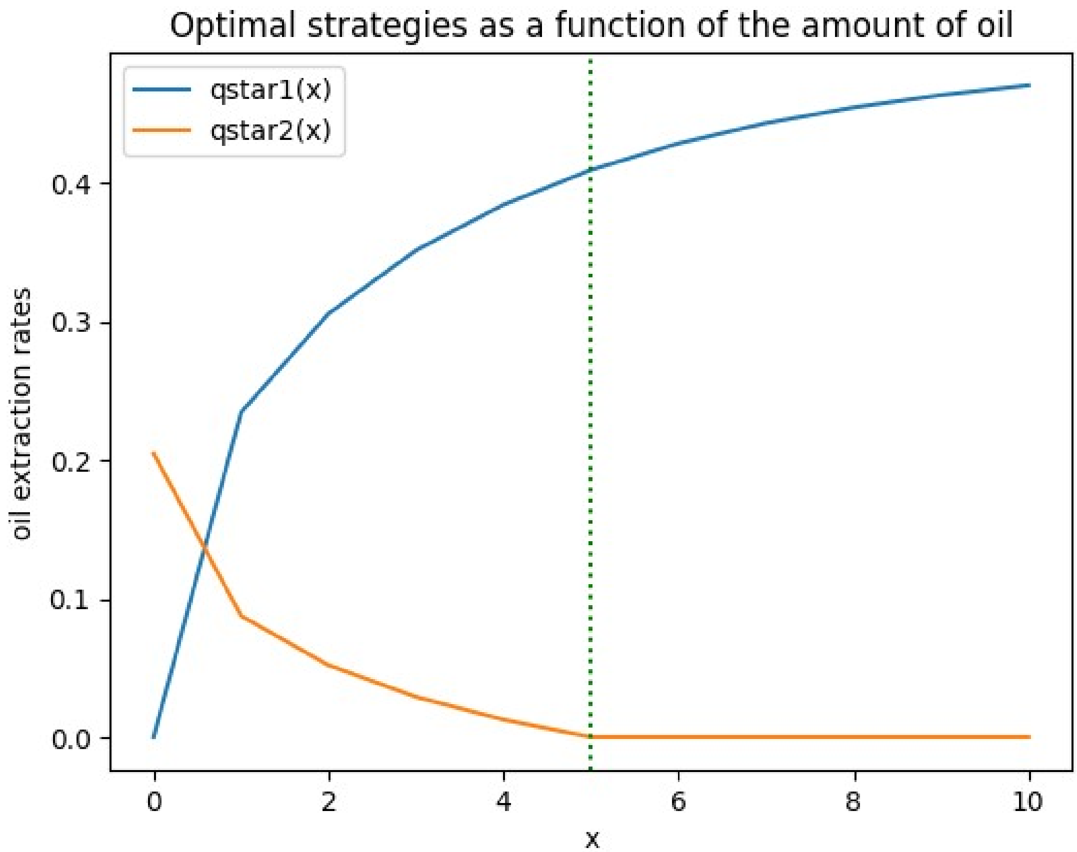
```
11.061252421053245
```

```
t2 = 1/r*math.log(mylambertw(theta_large(1, 0.1,
0)).real/mylambertw(theta_large(1, 0.1, 5)).real)
print(t2)
```

```
18.414056604369605
```

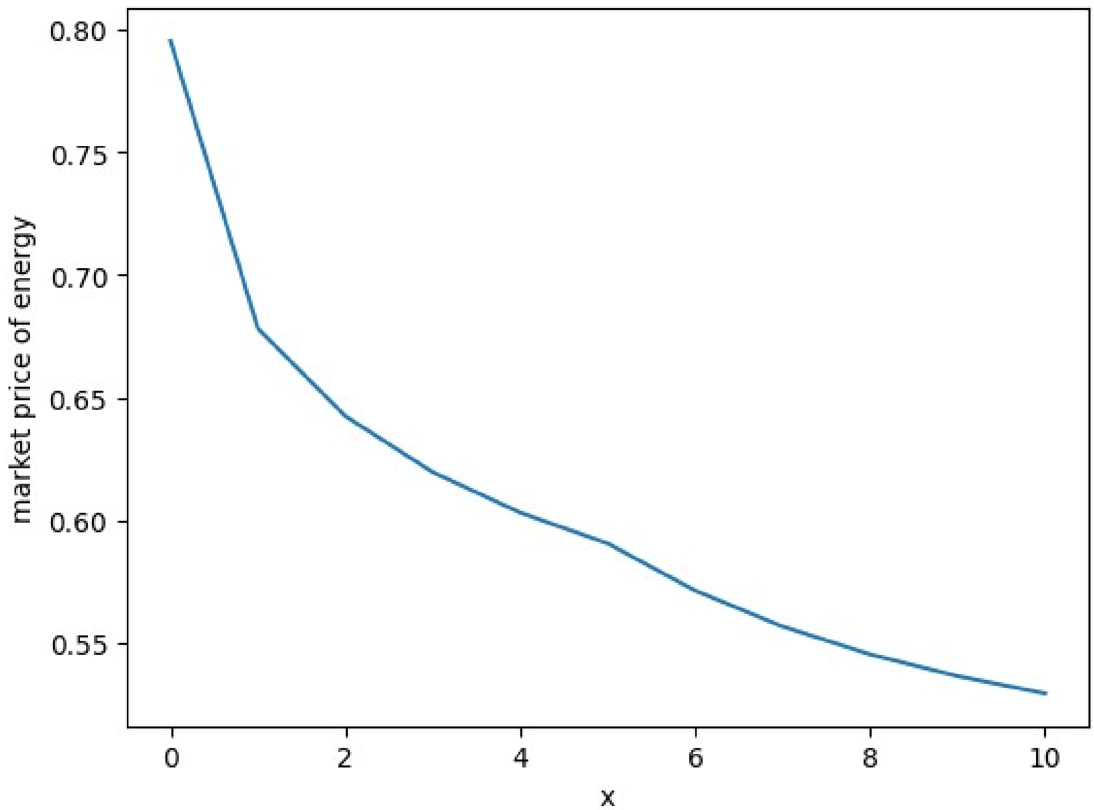
(d)

```
def qstar1(c, r, x, xb):
    if(x >= xb):
        return (1+mylambertw(theta_large(c, r, x-xb)).real)/2
    return (1+c)/3*(1+mylambertw(theta_small(c, r, x)).real)

def qstar2(c, r, x, xb):
    if(x >= xb):
        return 0
    return 1/3*(1-2*c-(1+c)/2*mylambertw(theta_small(c, r, x)).real)

xvec = [i for i in range(11)]
qstar1vec = []
qstar2vec = []
for x in xvec:
    qstar1vec.append(qstar1(c, r, x, xb))
    qstar2vec.append(qstar2(c, r, x, xb))
qstar1vec = np.array(qstar1vec)
qstar2vec = np.array(qstar2vec)
plt.plot(xvec, qstar1vec, label = "qstar1(x)")
plt.plot(xvec, qstar2vec, label = "qstar2(x)")
plt.legend()
plt.xlabel("x")
plt.ylabel("oil extraction rates")
plt.axvline(5, color = 'g', linestyle = "dotted")
plt.title("Optimal strategies as a function of the amount of oil")
plt.show()
```



(e)

```
plt.plot(xvec, 1-qstar1vec-qstar2vec)
plt.xlabel("x")
plt.ylabel("market price of energy")
# plt.title("Optimal strategies as a function of the amount of oil")
plt.show()
```



Q3

Ideas:

position of argmax

position of argmin

Last - First

Max - Min

Total slope/one slope (slope of one segment)

Sum of the absolute values of the slope:

- For C, B => ~equal to the actual start to end slope
- For Humped, twice or much larger
- For flat, almost equal to 0

Check for presence of 3-4 length decreasing/increasing subarray

- For humped max, increasing then decreasing; for humped min, vice versa
- 1 -> Contango
- 2 -> Backwardation
- 3 -> Humped peak

- 4 -> Humped trough
- 5 -> flat
- 6 -> Humped peak trough

Can or maybe should construct a table containing these signal values for each of the curves so as to determine which signals are the storngest/sharpest/best differentiators between curves and emply them while differentiating the curves.

```
ground_truth = pd.DataFrame(index = [2021, 2022, 2023, 2024, 2025],
columns = ["HG", "NG"])
ground_truth.loc[:, "HG"] = [[4], [1, 4], [1, 4], [1, 2, 3], [1]]
ground_truth.loc[:, "NG"] = [[2, 4, 6], [1, 2, 4, 6], [1, 3], [1, 3],
[2, 3]]
ground_truth
```

	HG	NG
2021	[4]	[2, 4, 6]
2022	[1, 4]	[1, 2, 4, 6]
2023	[1, 4]	[1, 3]
2024	[1, 2, 3]	[1, 3]
2025	[1]	[2, 3]

(a)

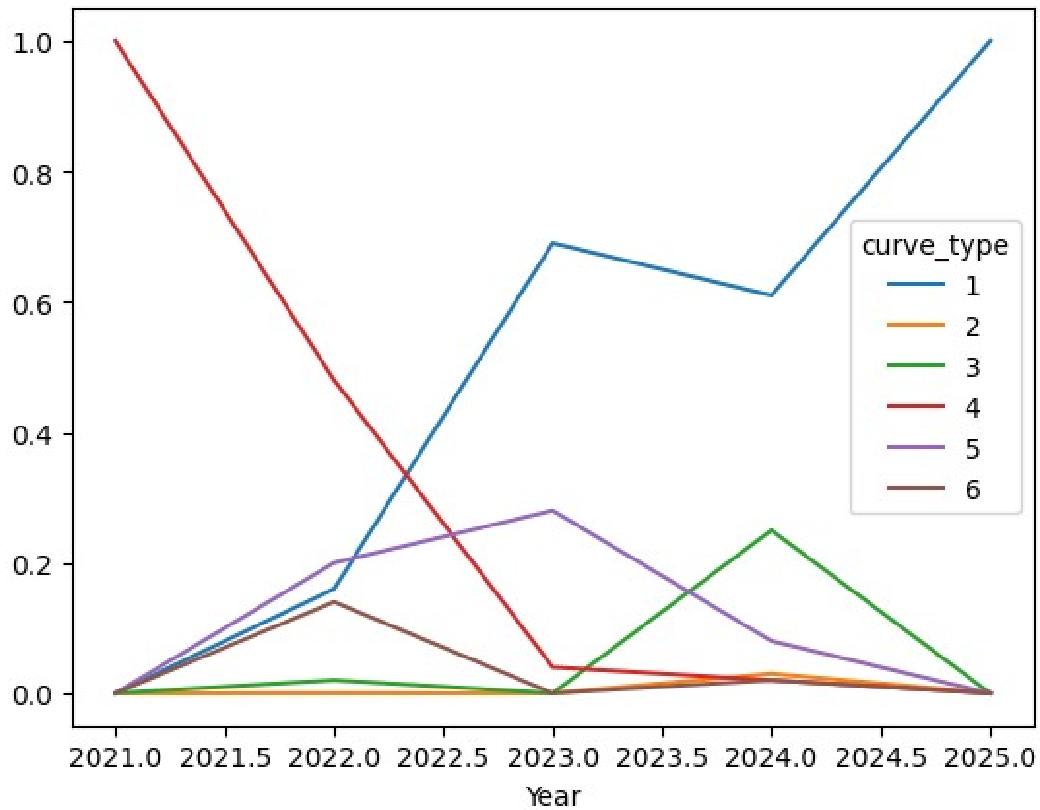
```
def determine_curve_type(v):
    lv = len(v)
    maxi = v.argmax() + 1
    mini = v.argmin() + 1
    lmf_frac = abs(v.iloc[-1] - v.iloc[0])/v.iloc[0]
    maxmf_frac = (v.iloc[maxi-1] - v.iloc[0])/v.iloc[0]
    minmf_frac = (v.iloc[0] - v.iloc[mini-1])/v.iloc[0]

    mark1 = lv/6
    mark2 = lv-lv/6

    # Contango: (similar structure for Backwardation)
    #   main signal: maxi in last 3-4, min in first 3-4
    #   lmf_frac: for filtering out noise
    if(mini <= mark1 and lmf_frac>=0.9*maxmf_frac):
        curve_type = 1
    # Backwardation: maxi in first 3-4, mini in last 3-4
    elif(maxi <= mark1 and lmf_frac>=0.9*minmf_frac):
        curve_type = 2
    # Humped peak trough
    elif(mark1<maxi<mark2 and mark1<mini<mark2):
        curve_type = 6
    # Humped peak: maxi in middle 8-10
    elif(mark1<maxi<mark2 and maxmf_frac>=0.01):
        curve_type = 3
    # Humped trough: mini in middle 8-10
```

```
    elif(mark1<mini<mark2 and minmf_frac>=0.01):
        curve_type = 4
    # flat
    else:
        curve_type = 5

    return curve_type

# Data fetching and filtering
## HG
data_hg = pd.read_excel("fcurves_data.xlsx", sheet_name = "HG")

## NG
data_ng = pd.read_excel("fcurves_data.xlsx", sheet_name = "NG")
date_col = data_ng[['Dates']]
# Select every 12th contract, starting from NG1
annual_contracts = data_ng.iloc[:, 1::12]
# Combine the date column with the selected contracts
data_ng = pd.concat([date_col, annual_contracts], axis=1)

# Data Processing
data = [data_hg, data_ng]
names = ["HG", "NG"]
num_dates, num_contracts = [], []
ldata = len(data)
for i in range(ldata):
    data[i] = data[i][data[i].iloc[:, 1:].isnull().all(axis=1) ==
False].reset_index(drop = True) # Keep those rows where not all values
are null (drop those where all values are null)
    # Feature generation
    # num_dates = data[i].shape[0]
    # num_contracts = data[i].shape[1] - 1 # since first column is of
Date
    num_dates.append(data[i].shape[0])
    num_contracts.append(data[i].shape[1] - 1) # since first column is
of Date
    data[i]["curve_type"] = data[i].apply(lambda x:
determine_curve_type(x.iloc[1:num_contracts[i]+1]), axis=1)
    data[i]["Year"] = data[i]["Dates"].transform(lambda x: x.year)

for i in range(ldata):
    table = data[i].groupby("curve_type").size()/len(data[i])*100
    print(names[i])
    print(table)
    print("")

HG
curve_type
1    54.554171
2     0.767018
```

```
3     6.327900
4    20.997124
5    13.518696
6     3.835091
dtype: float64

NG
curve_type
1    20.038351
2     9.971237
3    44.391179
4    19.654842
5     1.150527
6     4.793864
dtype: float64
```

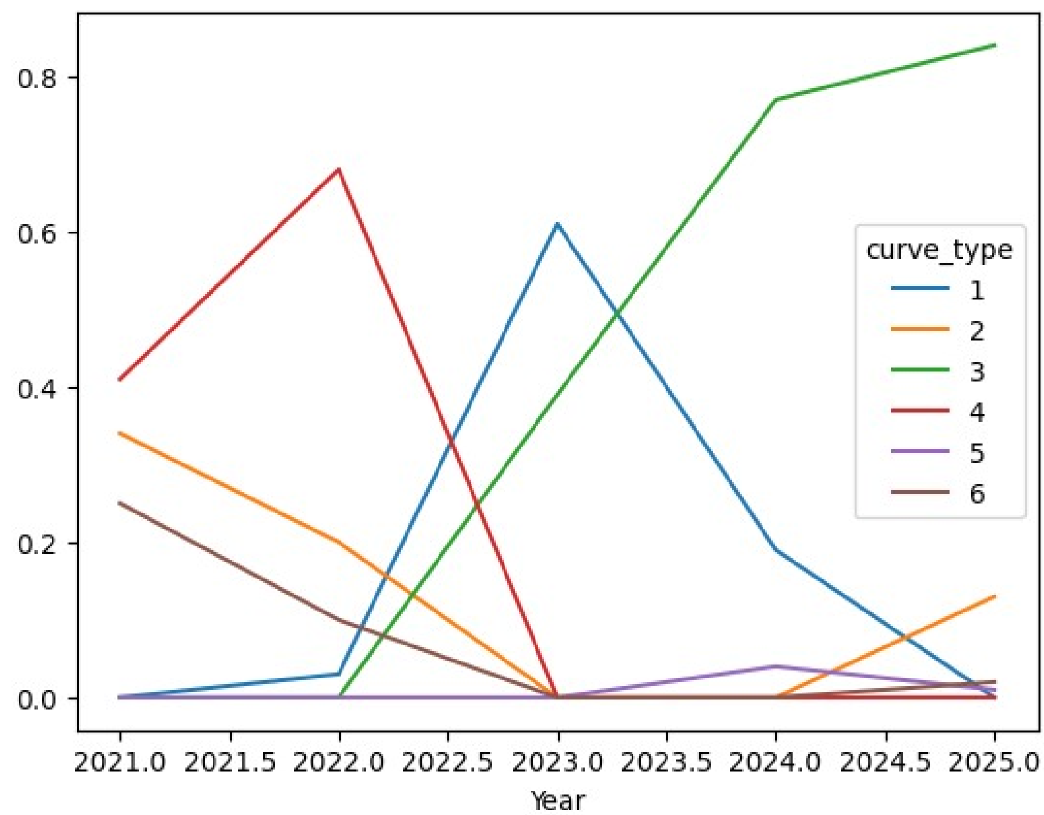
(b)

```
for i in range(ldata):
    data_by_year_curve = data[i].groupby(["Year",
"curve_type"]).size().unstack(fill_value = 0)
    data_by_year_curve =
data_by_year_curve.div(data_by_year_curve.sum(axis = 1),
axis=0).round(2)
    print(names[i])
    print(data_by_year_curve)
    print("")
    data_by_year_curve.plot()
    plt.show()
# type(data_hg_by_year_curve) # pd.DataFrame

HG
curve_type     1     2     3     4     5     6
Year
2021        0.00  0.00  0.00  1.00  0.00  0.00
2022        0.16  0.00  0.02  0.48  0.20  0.14
2023        0.69  0.00  0.00  0.04  0.28  0.00
2024        0.61  0.03  0.25  0.02  0.08  0.02
2025        1.00  0.00  0.00  0.00  0.00  0.00
```

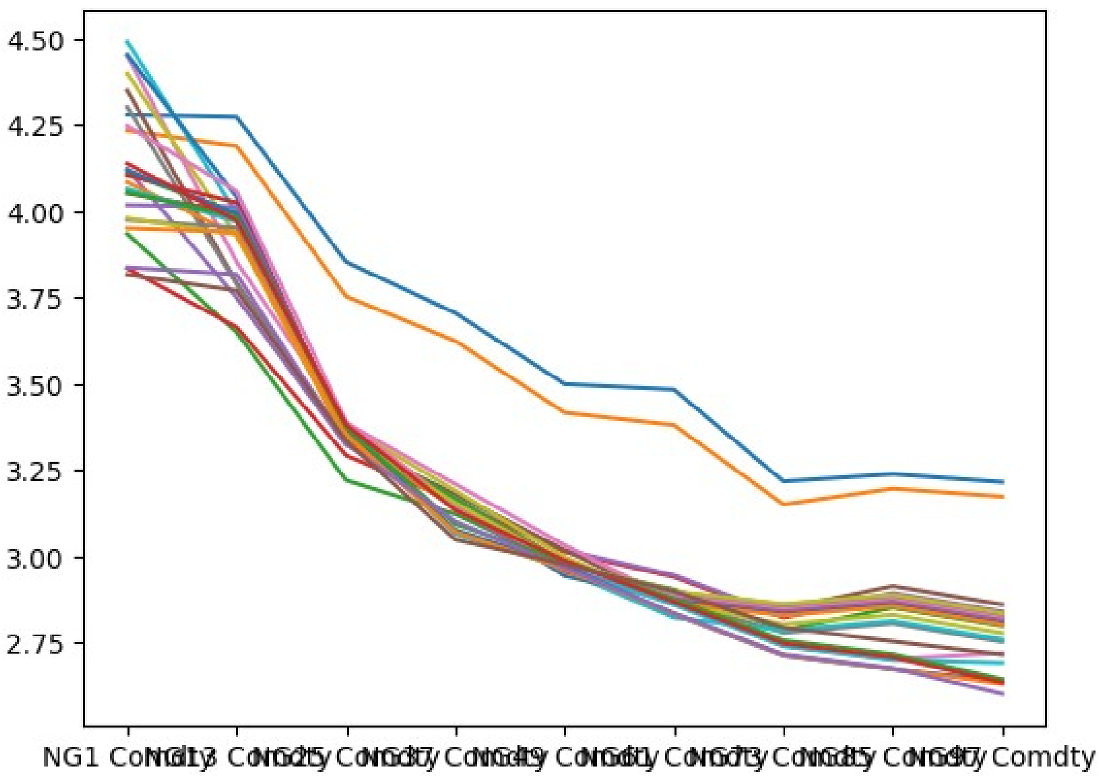


NG

curve_type	1	2	3	4	5	6
Year						
2021	0.00	0.34	0.00	0.41	0.00	0.25
2022	0.03	0.20	0.00	0.68	0.00	0.10
2023	0.61	0.00	0.39	0.00	0.00	0.00
2024	0.19	0.00	0.77	0.00	0.04	0.00
2025	0.00	0.13	0.84	0.00	0.01	0.02




```
data_index = 1 #  & (data[data_index]["curve_type"]==4)
temp = data[data_index][(data[data_index]["Year"] == 2025) &
(data[data_index]["curve_type"] == 2)]
temp.iloc[:, 1: num_contracts[data_index]+1].T.plot(legend = False)
# temp.iloc[np.random.randint(0, len(temp)),
1:num_contracts[data_index]+1].T.plot(legend = False)
# data[data_index][(data[data_index]["Year"] == 2022) &
(data[data_index]["curve_type"] == 2)].iloc[48,
1:num_contracts[data_index]+1].T.plot(legend = False)

# Problems:
## HG-2021:
## HG-2022: 5 should be 1 (probably because min index is in between)
(false negative)
## HG-2023: 5 should be 4/6 (false neg)
## HG-2024: 5 should be 3
## HG-2025:
## NG-2021:
## NG-2022:
## NG-2023:
## NG-2024:
## NG-2025:

<Axes: >
```



(c)

HG

- Humped troughs from 2021-2023 followed by Contangos (2023-2025)
- One of the main geopolitical events was the divergence between COMEX and LME due to imposition of US tariffs; this led to higher prices in COMEX (price premiums) as compared to LME where the prices were more subdued. Thus, the backwardations and humped troughs (in the initial years like 2021-2023) changed to strong contangos in late 2024 and 2025 (after/around when the tariffs were introduced and the tariff war began).

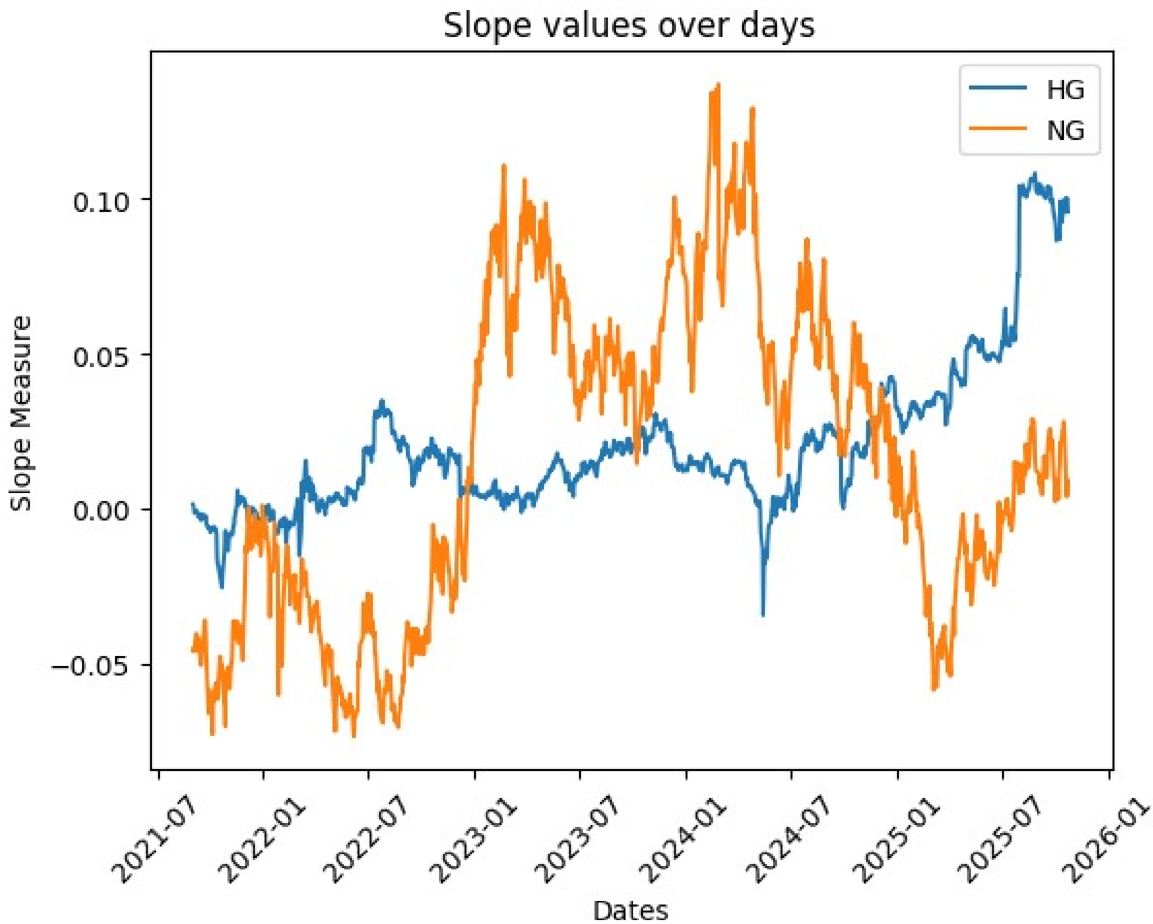
NG

- In 2021-2022, the curve was considerably downward slopping (either in complete backwardation or if not, then as a humped trough or humped peak trough)
- 2023-2024 saw the curve change to increasing (as contango (2023) or as humped peak (2024, 5))
- Yes, there were some structural changes that led the Backwardation seen in 2021-22 change to Contango in 2023-24 and eventually to tighter and flat prices in/around 2025
- Around 2021-22, there was increase in demand in the short term. What was seen was immediate supply shortages and high demand. Why? the number of drilled wells (supplying NG) got limited (probably due to COVID); thus US ability to supply NG was limited due to which supply fell short of demand.
- Then in 2023-24, as supply increased, storage levels built up and supply surplus remained high, short term prices went down, thus triggering contango.

(d)

- Slope is a measure of the rate of change of y with respect to x (i.e. dy/dx). In discrete terms, it can be approximated as delta_y/delta_x. However, this generally is done when the curve is smooth. In our case, it is not. So, we take average of the slopes of each piece (note that our function is piecewise linear) and define this as the slope for that day.
- However, also note that scales (of price change) of HG and NG are different. Hence, to compare across products/construct a robust measure, it is better to convert numbers to percentages.
- Also note that for NG, 2 points are 1 year apart whereas for HG, they are just 1 month apart. So, take that also into account.
- If we take the absolute values of the consecutive point differences, that parameter will probably be higher for humped curves than for contango and backwardation; this absolute value parameter will capture the volatility of the curve.

```
def average_slope(v, dnm):
    lv = len(v)
    val = 0
    for i in range(lv-1):
        val += ((v.iloc[i+1]-v.iloc[i])/v.iloc[i])/dnm # normal sum
        # val += abs(v.iloc[i+1]-v.iloc[i])/v.iloc[i] # absolute sum
    val /= (lv-1)
    return val

period_coefficient = [1/12, 1]
for i in range(ldata):
    data[i]["slope_value"] = data[i].apply(lambda x:
average_slope(x[1:num_contracts[i]+1], period_coefficient[i]), axis=1)
    plt.plot(data[i]["Dates"], data[i]["slope_value"], label =
names[i])
plt.xticks(rotation = 45)
plt.xlabel("Dates")
plt.ylabel("Slope Measure")
plt.legend()
plt.title("Slope values over days")
plt.show()
```



Q4

PJM

Date: 2025-10-31, Time: 12:57 PM ET

(a)

- Electricity Demand: 87105 MW
- Met by Renewables: 13426 MW

```
print("Fraction of demand met by renewables:", 13426/87105)

Fraction of demand met by renewables: 0.15413581309913324
```

(b)

```
fuel_generated = {}
fuel_generated["Coal"] = 11842
fuel_generated["Gas"] = 33381
fuel_generated["Hydro"] = 344
```

```
fuel_generated["Multiple Fuels"] = 1002
fuel_generated["Nuclear"] = 27279
fuel_generated["Oil"] = 175
fuel_generated["Other Renewables"] = 527
fuel_generated["Solar"] = 8606
fuel_generated["Wind"] = 3949
fuel_generated = {x: v*100/sum(fuel_generated.values()) for x, v in 
fuel_generated.items()}
table = pd.DataFrame(sorted(fuel_generated.items(), key = lambda x: -
x[1]), columns = ["Fuel", "% supplied"]).round(2)
table
```

	Fuel	% supplied
0	Gas	38.32
1	Nuclear	31.32
2	Coal	13.60
3	Solar	9.88
4	Wind	4.53
5	Multiple Fuels	1.15
6	Other Renewables	0.61
7	Hydro	0.39
8	Oil	0.20

(c)

Actuals (represented by Generation plot on the website) have mostly been able to keep track of the Forecast (represented by Forecast plot on the website) from 12 am to 12 pm except between 3:30 to 7:30 AM where the Actuals were less than the Forecast (although by a small fraction~3-5%).

As far as day-ahead forecasts are concerned, the actuals (or originals) are 5-7K more than the day-ahead forecasts (referred from dataviewer.pjm.com).

(d)

- Highest LMP: 37.08 at PEPCO (Washington DC area)
- Lowest LMP: 20.48 at EKPC (East Kentucky)

Why is PEPCO High?

1. High demand from data centers, industrial facilities, extreme weather and urban and city population
2. PEPCO depends mainly on the non-renewable sources of energy (Gas, Nuclear, etc) which are now retiring. It did not accept proposals for Renewable energies like Solar and Wind back in 2021 due to which they just contribute 4% of the total supply.
3. High Delivery Fees, High taxes add up to the cost.

Why is EKPC low?

1. It serves mainly to the rural and sub-urban areas where demand for the electricity is not so high.

2. EKPC is based on coal fuel (Mason, Pulaski; Kentucky being the 6th largest coal producing state, so lots of coal resources => not retiring) and Natural gas plants (Clark, Oldham) that are still abundant along with renewable energy resourses (Baron, Greenup). Multiple sources of energy supply keep prices low.
3. EKPC is a not-for-profit cooperative organization and hence, would strive to keep the prices low.

ERCOT

Date: 2025-10-31, Time: 1:27 PM ET

(a)

- Current Demand: 50094
- fraction fulfilled by Renewables = 64.5% ~ 0.645

(b)

```
fuel_generated = {}
fuel_generated["Coal and Lignite"] = 4287
fuel_generated["Hydro"] = 0
fuel_generated["Natural Gas"] = 9925
fuel_generated["Nuclear"] = 3738
fuel_generated["Other"] = 0
fuel_generated["Power Storage"] = 43
fuel_generated["Solar"] = 25673
fuel_generated["Wind"] = 7008
fuel_generated = {x: v*100/sum(fuel_generated.values()) for x, v in fuel_generated.items()}
table = pd.DataFrame(sorted(fuel_generated.items(), key = lambda x: -x[1]), columns = ["Fuel", "% supplied"]).round(2)
table
```

	Fuel	% supplied
0	Solar	50.66
1	Natural Gas	19.59
2	Wind	13.83
3	Coal and Lignite	8.46
4	Nuclear	7.38
5	Power Storage	0.08
6	Hydro	0.00
7	Other	0.00

(c)

Here, by forecast, we mean the demand. For the previous hours, the committed capacity (or the supply) is considerably more than the demand to be fulfilled (Almost 14k MW more => 30-35% more than the demand). For day-ahead forecasts, the supply is again well above the forecasts (for Nov 1-2-3-4-5) i.e. around 1.4-1.8 times that of the forecast.

(d)

Price is highest around the NorthCentral/East Region border; whereas it is the lowest diametrically opposite (farwest/west regions' border).

Northcentral/East Texas

- Dependent solely on ERCOT which is cut off/isolated from Eastern or Western US electricity interconnections, thus leading to kind of monopoly and hence higher costs
- more hot and tougher summers leading to more energy demand
- Consists of major population and industrial centers like Dallas and Fort Worth, leading to higher demand

West Texas

- More remote and rural, leading to modest demand
- More scope and investment in solar and wind energy resources (due to great plains present in/extending to the west texas region) leading to duopoly and stability of prices (Eg: Roscoe: largest wind farm of the state)
- Some areas of far west texas extend into/are supported by the western electricity interconnections (apart from ERCOT) leading to a more stable supply of electricity during ciritical times

References:

- https://www.ecoflow.com/us/blog/texas-electricity-prices-solar-solutions